

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (Currently amended) A method for managing heap memory in a
2 multitasking system, comprising:
3 reserving a guaranteed amount of heap memory for a task from a common
4 heap in the multitasking system, wherein the heap memory reserved for the task is
5 separate from heap memory reserved for all other tasks in the common heap of the
6 multitasking system;
7 receiving a request from the task to allocate heap memory for a new
8 object; and
9 if heap memory is available in the guaranteed amount of heap memory for
10 the task,
11 allocating heap memory for the new object from the guaranteed amount of
12 heap memory;
13 if not, checking if surplus heap memory is available in the common heap;
14 if so, reserving an additional amount of heap memory to the task from the
15 common heap and allocating heap memory for the new object from the additional
16 amount of heap memory, whereby allocating heap memory for the new object
17 from the additional amount of heap memory delays garbage collection; and
18 if not, performing garbage collection on the heap memory reserved for the
19 task, wherein the task space is separate from all other task space, whereby other
20 tasks continue normal execution without interruption due to garbage collection, or
21 subsequent memory compaction.

22 otherwise, performing one of the following:
23 garbage collection on the heap memory reserved for the
24 task, wherein the task space is separate from all other task space, whereby other
25 tasks do not need to be paused during garbage collection; and
26 if surplus heap memory is available in the common heap in
27 addition to heap memory allocated to tasks,
28 reserving an additional amount of heap memory to
29 the task from the common heap, and
30 allocating heap memory for the new object from the
31 additional amount of heap memory, whereby allocating heap
32 memory for the new object from the additional amount of heap
33 memory delays garbage collection.

1 2. (Previously presented) The method of claim 1, wherein if surplus heap
2 memory is not available in the common heap in addition to heap memory
3 allocated to tasks, the method further comprises
4 performing garbage collection on heap memory to reclaim unused reserved
5 heap memory, and
6 allocating heap memory for the new object from reclaimed surplus heap
7 memory.

1 3. (Previously presented) The method of claim 1, wherein reserving the
2 guaranteed amount of heap memory from the common heap includes:
3 determining if there is sufficient heap memory available in the common
4 heap; and
5 if not, performing garbage collection to reclaim allocated surplus heap
6 memory, and
7 reserving heap memory for the task from reclaimed heap memory.

1 4. (Previously presented) The method of claim 1, wherein heap memory in
2 the common heap is managed using a generational garbage collector.

1 5. (Original) The method of claim 4, wherein a plurality of tasks share an
2 old generation of the generational garbage collector.

1 6. (Original) The method of claim 5, wherein each task of the plurality of
2 tasks has a new generation of the generational garbage collector belonging to the
3 task.

1 7. (Original) The method of claim 4, wherein the generational garbage
2 collector is a copying garbage collector.

1 8. (Currently amended) A computer-readable storage medium storing
2 instructions that when executed by a computer cause the computer to perform a
3 method for managing heap memory in a multitasking system, the method
4 comprising:

5 reserving a guaranteed amount of heap memory for a task from a common
6 heap in the multitasking system, wherein the heap memory reserved for the task is
7 separate from heap memory reserved for all other tasks in the common heap of the
8 multitasking system;

9 receiving a request from the task to allocate heap memory for a new
10 object; and

11 if heap memory is available in the guaranteed amount of heap memory for
12 the task,

13 allocating heap memory for the new object from the guaranteed amount of
14 heap memory;

15 | if not, checking if surplus heap memory is available in the common heap;

16 if so, reserving an additional amount of heap memory to the task from the
17 common heap and allocating heap memory for the new object from the additional
18 amount of heap memory, whereby allocating heap memory for the new object
19 from the additional amount of heap memory delays garbage collection; and
20 if not, performing garbage collection on the heap memory reserved for the
21 task, wherein the task space is separate from all other task space, whereby other
22 tasks continue normal execution without interruption due to garbage collection, or
23 subsequent memory compaction.
24 otherwise, performing one of the following:
25 garbage collection on the heap memory reserved for the
26 task, wherein the task space is separate from all other task space, whereby other
27 tasks do not need to be paused during garbage collection; and
28 if surplus heap memory is available in the common heap in
29 addition to heap memory allocated to tasks,
30 reserving an additional amount of heap memory to
31 the task from the common heap, and
32 allocating heap memory for the new object from the
33 additional amount of heap memory, whereby allocating heap
34 memory for the new object from the additional amount of heap
35 memory delays garbage collection.

1 9. (Previously presented) The computer-readable storage medium of claim
2 8, wherein if surplus heap memory is not available in the common heap in
3 addition to heap memory allocated to tasks, the method further comprises:
4 performing garbage collection on heap memory to reclaim unused reserved
5 heap memory, and
6 allocating heap memory for the new object from reclaimed surplus heap
7 memory.

1 10. (Previously presented) The computer-readable storage medium of
2 claim 8, wherein reserving the guaranteed amount of heap memory from the
3 common heap includes:
4 determining if there is sufficient heap memory available in the common
5 heap; and
6 if not, performing garbage collection to reclaim allocated surplus heap
7 memory, and
8 reserving heap memory for the task from reclaimed heap memory.

1 11. (Previously presented) The computer-readable storage medium of
2 claim 8, wherein heap memory in the common heap is managed using a
3 generational garbage collector.

1 12. (Original) The computer-readable storage medium of claim 11,
2 wherein a plurality of tasks share an old generation of the generational garbage
3 collector.

1 13. (Original) The computer-readable storage medium of claim 12,
2 wherein each task of the plurality of tasks has a new generation of the generational
3 garbage collector belonging to the task.

1 14. (Original) The computer-readable storage medium of claim 11,
2 wherein the generational garbage collector is a copying garbage collector.

1 15. (Currently amended) An apparatus that facilitates managing computer
2 heap memory in a multitasking system, comprising:
3 a computing device including a multitasking virtual machine;

4 a reserving mechanism within the multitasking virtual machine that is
5 configured to reserve a guaranteed amount of physical heap memory for a task
6 from a new generation space within a common heap in the multitasking system,
7 wherein the heap memory reserved for the task is separate from heap memory
8 reserved for all other tasks in the common heap of the multitasking system;

9 a receiving mechanism within the multitasking virtual machine that is
10 configured to receive a request from the task to allocate heap memory for a new
11 object;

12 the reserving mechanism that is further configured to reserve an additional
13 amount of heap memory to the task from the common heap; and

14 an allocating mechanism within the multitasking virtual machine that is
15 configured to allocate heap memory for the new object from the guaranteed
16 amount of heap memory

17 a reserving mechanism configured to reserve an additional amount of heap
18 memory to the task from the common heap and allocating heap memory for the
19 new object from the additional amount of heap memory, whereby allocating heap
20 memory for the new object from the additional amount of heap memory delays
21 garbage collection; and

22 a garbage collection mechanism configured to perform garbage collection
23 on the heap memory reserved for the task, wherein the task space is separate from
24 all other task space, whereby other tasks continue normal execution without
25 interruption due to garbage collection, or subsequent memory compaction.

1 16. (Previously presented) The apparatus of claim 15, further comprising:
2 a garbage collecting mechanism that is configured to perform a garbage
3 collection on heap memory to reclaim unused reserved heap memory, and
4 the allocating mechanism that is further configured to allocate heap
5 memory for the new object from reclaimed heap memory

1 17. (Previously presented) The apparatus of claim 15, further comprising:
2 a determining mechanism that is configured to determine if there is
3 sufficient heap memory available in the common heap; and
4 a garbage collection mechanism that is configured to perform a garbage
5 collection to reclaim allocated surplus heap memory, and
6 the reserving mechanism that is further configured to reserve heap memory
7 for the task from reclaimed heap memory.

1 18. (Original) The apparatus of claim 15, further comprising a generational
2 garbage collector.

1 19. (Original) The apparatus of claim 18, wherein a plurality of tasks share
2 an old generation of the generational garbage collector.

1 20. (Original) The apparatus of claim 19, wherein each task of the plurality
2 of tasks has a new generation of the generational garbage collector belonging to
3 the task.

1 21. (Original) The apparatus of claim 18, wherein the generational garbage
2 collector is a copying garbage collector.